**1) What Is Ansible?**

Ansible is a configuration management system. It is used to set up and manage infrastructure and applications. It allows users to deploy and update applications using SSH, without needing to install an agent on a remote system.

**2) What’s the use of Ansible?**

Ansible is used for managing IT infrastructure and deploy software apps to remote nodes.

For example, Ansible allows you to deploy as an application to many nodes with one single command. However, for that, there is a need for some programming knowledge to understand the ansible scripts.

**3) What is Ansible Galaxy?**

Ansible can communicate with configured clients from the command line by using ansible command. It also allows you to automate configuration by using ansible-playbook command. To create the base directory structure, you can use a tool bundled with Ansible which is known as ansible-galaxy.

**Command:**



|  |  |
| --- | --- |
| 1 | $ ansible-galaxy init azavea. packer |

azavea.packer was created successfully

**4) What is Continuous Delivery?**

Continuous delivery is a practice of delivering the software as soon as it developed. In this method, we need to use versioning control system. The software is constantly updated in live production systems.

**5) What is the way to access shell environment variables in Ansible?**

In Ansible, if you want to access existing variables, the user needs to use the ‘env’ lookup plugin. Example, to access the value of the Office environment on the management machine:

You need to write following code:



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | ---  # ...  vars:  local\_home: "{{ lookup('env','Office') }}"  I  {{ ansible\_env.SOME\_VARIABLE }} |

**6) What is the code you need to write for accessing a variable name?**

Variable names can be built by adding using the following method:



|  |  |
| --- | --- |
| 1 | {{ hostvars[inventory\_hostname]['ansible\_' + which\_interface]['ipv4']['address'] }} |

The method of using hostvars is important because it’s a dictionary of the entire namespace of variables. ‘inventory\_hostname’ variable specifies the current host you are looking over in the host loop.

**7) Explain how you can disable cowsay?**

If cowsay is installed then executing playbooks inside the Ansible you can disable coway by using following  options:

1. Uninstall cowsay
2. Setting up value for the environment variable

export ANSIBLE\_NOCOWS=1

**8) Explain how you can copy file recursively onto a target host?**

The “copy” module has a recursive parameter. However, if you want this to perform more efficient for a large number of files, then “synchronize” module is the best option for you.

**9) How Can you submit a change to the Documentation in Ansible?**

Documentation for Ansible is kept in the project git repository. It contains complete instructions for contributing can be found in the docs.

**10) What Is the Best Method to Make Content Reusable/redistributable?**

You can read everything about “Roles” in the playbooks documentation section. This helps to make playbook content self-contained and shareable with other ansible users.

**11) What is Ansible Tower?**

Ansible tower is a tool which makes Ansible very easy to use. It acts as a hub for the task automation. The tower is free for usage till 10 nodes.

**12) What’s the method to check the inventory vars defined for the host?**

**For that use this command:**



|  |  |
| --- | --- |
| 1 | ansible -m debug -a "var=hostvars['hostname']" localhost |

**13) State the difference between Variable name and Environment Variables.**

|  |  |
| --- | --- |
| **Variable Name** | **Environment Variables** |
| It can be built by adding strings. | To access the environment variable, you need to access existing variables. |
| {{ hostvars[inventory\_hostname][‘ansible\_’ + which\_interface][‘ipv4’][‘address’] }} | # … vars: local\_home: “{{ lookup(‘env’,’HOME’) }}” |
| Allows to add strings | To set environment variables, we need to see the advanced playbooks section. |
| Ipv4 address type use for Variable names we use the ipv4 address. | For Remote environment variables, use {{ ansible\_env.SOME\_VARIABLE }} |

**14) What are ad-hoc commands?**

You can think of ad-hoc commands as a way for us to take actions on our hosts without writing a playbook. For example, if we want to reboot all hosts in a particular group(webservers). Then you can write a playbook or simply run a one-off ad-hoc command.

**15) Explain Ansible facts**

You can think of ansible facts as a way for ansible to get information about a host and store them in variables for easy access. This information stored in predefined variables are available to use in the playbook. To generate facts, ansible runs the setup module.

**16) How do you see all variables for a host?**

You can see them using the host vary variable. This stores host variables with the hostname as key. For example, to look at the variables defined for localhost, you can run;



|  |  |
| --- | --- |
| 1 | ansible -m debug -a "var=hostvars[inventory\_hostnam |

**17) Explain modules in ansible**

Modules in Ansible are idempotent. From a RESTful service standpoint, for the operation to be idempotent, clients can perform the same result by using modules in Ansible. Multiple identical requests become a single request.

There are two different types of modules in Ansible:

* **Core modules**
* **Extras modules**

**Core Modules**

The Ansible team maintains these types of modules, and they will always ship with Ansible software. They will also give higher priority for all requests than those in the “extras” repos.

**Extras Modules:**

These modules currently is bundled with Ansible but might available separately in the future. They are also mostly maintained by the Ansible community. These modules are still usable, but it can receive a lower rate of response to issues and pull requests.

**18) When should you test playbooks and roles?**

In ansible, Tests can be added either in new Playbooks or to existing Playbooks. Therefore, most of the testing job offers a clean hosting each time. By using this testing methodology, you need to make very little to no code changes.

**19) Discuss method to Create an Empty File with Ansible**

**To create and empty file you need to follow given steps.**

Step 1. Save An Empty File into The Files Directory

Step 2. Copy It to The Remote Host.

**Q. What is Ansible?**  
Ansible is developed in Python language.  
It is a software tool. It is useful while deploying any application using ssh without any downtime. Using this tool one can manage and configure software applications very easily.

**Q. Ansible Playbooks vs Roles**

|  |  |
| --- | --- |
| **Roles** | **Playbooks** |
| Roles are reusable subsets of a play. | Playbooks contain Plays. |
| A set of tasks for accomplishing certain role. | Mapps among hosts and roles. |
| Example: common, webservers. | Example: site.yml, fooservers.yml, webservers.yml. |

**Q: What are the advantages of using Ansible?**

The main three advantages of using this tool is,i.e. Ansible

* Agentless
* Very low overhead
* Good performance

**Q. Comapre Ansible VS Puppet**

|  |  |
| --- | --- |
| **Ansible** | **Puppet** |
| Simplest Technology | Complex Technology |
| Written in YAML language | Written in Ruby language |
| Automated workflow for Continuous Delivery | Visualization and reporting |
| Agent-less install and deploy | Easy install |
| No support for Windows | Support for all major OS’s |
| Good GUI | GUI - work under progress |
| CLI accepts commands in almost any language | Must learn the Puppet DSL |

**Q. How Ansible Works?**  
There are many similar automation tools available like [Puppet](https://mindmajix.com/puppet-training), Capistrano, Chef, Salt, Space Walk etc, but Ansible categorize into two types of server: controlling machines and nodes.

The controlling machine, where Ansible is installed and Nodes are managed by this controlling machine over SSH. The location of nodes are specified by controlling machine through its inventory.

The controlling machine (Ansible) deploys modules to nodes using SSH protocol and these modules are stored temporarily on remote nodes and communicate with the Ansible machine through a JSON connection over the standard output.

Ansible is agent-less, that means no need of any agent installation on remote nodes, so it means there are no any background daemons or programs are executing for Ansible, when it’s not managing any nodes.

Ansible can handle 100’s of nodes from a single system over SSH connection and the entire operation can be handled and executed by one single command ‘ansible’. But, in some cases, where you required to execute multiple commands for a deployment, here we can build playbooks.  
Playbooks are bunch of commands which can perform multiple tasks and each playbooks are in YAML file format.

**Q.What’s the Use of Ansible.**  
Ansible can be used in IT Infrastructure to manage and deploy [software applications](https://mindmajix.com/software-application-development-courses) to remote nodes. For example, let’s say you need to deploy a single software or multiple software to 100’s of nodes by a single command, here ansible comes into picture, with the help of Ansible you can deploy as many as applications to many nodes with one single command, but you must have a little programming knowledge for understanding the ansible scripts.

We’ve compiled a series on Ansible, title ‘Preparation for the Deployment of your IT Infrastructure with Ansible IT Automation Tool‘, through parts 1-4 and covers the following topics.

**Q. How would you describe yourself in terms of what you do and how you’d like to be remembered?**  
Obviously I’d like to be remembered as a master of prose who forever changed the face of literature as we know it, but I’m going to have to settle for being remembered as a science fiction writer (and, more and more, critic) who wrote the occasional funny line and picked up a few awards.

**Q. Why are you attracted to science and science fiction?**  
Early imprinting, maybe, for the science fiction. When I was quite small a family friend let me read his 1950s run of ‘Galaxy’ magazine. My favourite aunt pressed John Wyndham’s ‘The Day of the Triffids’ on me; a more terrifying great-aunt gave me G.K. Chesterton’s fantastic novels; and so on.  
The incurable addiction had begun. Meanwhile, science classes just seemed to be the part of school that made most sense, and I fell in love with Pelican pop-maths titles – especially Kasner’s and Newman’s ‘Mathematics and the Imagination’ and all those books of Martin Gardner’s ‘Scientific American’ columns.

**Q. Tell us about your software company and what sort of software it produced(s).**  
This goes back to the 1980s and the Apricot home computers, the early, pretty and non-PC-compatible ones. My pal Chris Priest and I both used them for word processing, and he persuaded me to put together a disk of utilities to improve the bundled ‘SuperWriter’ w/p, mostly written in Borland Turbo Pascal 3 and later 4: two-column printing, automated book index preparation, cleaning the crap out of the spellcheck dictionary, patching SuperWriter to produce dates in UK format, and so on.

Then I redid the index software (‘AnsibleIndex’) in CP/M for the Amstrad PCW and its Locoscript word processors. When the Apricot market collapsed, I wrote an Apricot emulator in assembler so that people could keep using their horrible but familiar old software on a PC. Eventually, in a fit of nostalgia, I collected all my columns for ‘Apricot File’ and various Amstrad PCW magazines as books unoriginally titled ‘The Apricot Files’ and ‘The Limbo Files’. (That’s probably enough self-promotion, but there’s lots more at HTTP://ANSIBLE.CO.UK/.)

**Q. Describe your newsletter Ansible and who it’s aimed at.**  
It appears monthly and has been called the ‘Private Eye’ of science fiction, but isn’t as cruel and doesn’t (I hope) recycle old jokes quite as relentlessly. Though I feel a certain duty to list some bread-and-butter material like conventions, award winners and deaths in the field, ‘Ansible’ skips the most boring SF news – the long lists of books acquired, books published, book sales figures, major new remainders – in favour of quirkier items and poking fun at SF notables. The most popular departments quote terrible lines from published SF/fantasy and bizarre things said about SF by outsiders (‘As Others See Us’). All the back issues of ‘Ansible’ since it started in 1979 can be read online.

**Q: So how does Ansible work? Please explain in detail?**  
Within the market, they are many automation tools like Puppet, Capistrano, Chef, Salt, Space Walk etc.

When it comes to Ansible, this tool is categorized into two types of servers:  
1. Controlling machines  
2. Nodes

Ansible is an agentless tool so it doesn’t require any mandatory installations on remote nodes. So there is no background programs that are executed while it is managing any nodes.  
Ansible is able to handle a lot of nodes from a single system over SSH connection.  
Playbooks are defined as a bunch of commands where they are capable of performing multiple tasks and they are in YAML file format.

**Q: Do we have any Web Interface/ Rest API etc for this?**  
Yes, Ansible Inc makes a great efficient tool. It is easy to use.

**Q: What is Ansible Tower?**  
Ansible is classified as a web-based solution which makes Ansible very easy to use. It is considered to be or acts like a hub for all of your automation tasks. The tower is free for usage till 10 nodes.

**Q: How do change the documentation and submit it?**  
Usually, the documentation is kept in the main project folder in the git repository.  
For complete instructions on this can be available in docs.

**Q: How do you access Shell Environment Variables?**  
If you are just looking to access the existing variables then you can use “env” lookup plugin.  
For example:  
Accessing the value of Home environment variable on management machine:

local\_home:”{{lookup(‘env’,’HOME’)}}”

**Q: How can you speed up management inside in EC2?**  
It is not advised to manage a group of EC2 machines from your laptop.  
The best way is to connect to a management node inside Ec2 first and then execute Ansible from there.

**Q: How can you disable Cowsay?**  
If Cowsay is installed then executing your playbooks within Ansible is very smooth.  
Even if you think that you want to work in a professional cow free environment, then you will have two options:  
1.  Uninstall cowsay  
2. Setting up value for the environment variable, like below

Export ANSIBLE\_NOCOWS=1

**Q: How can you access a list of Ansible\_Variables?**  
By default, Ansible gathers facts under machines under management. Further, these facts are accessed in Playbooks and in templates. One of the best ways to view a list of all the facts that are available in a machine, then you need to run the setup module in the ad-hoc way:

Ansible- m setup hostname

Once this statement is executed, it will print out a dictionary of all the facts that are available for that particular host. This is the best way to access the list of Ansible\_variables.

**Q: How can you see all the inventory variables that are defined in the host?**  
The best way to see all the inventory variables is by executing this command below:

Ansible - m debug- a “var=hostvars[‘hostname’]” localhost

**Q: Why don’t you ship in X format?**  
They are several reasons for not shipping in X format. In general, it caters towards maintainability. Within the market, they are tons of different ways to ship software and it is very tedious to support all of them.

[Check Out Ansible Tutorials](https://mindmajix.com/ansible)

**Q: What is that Ansible can do?**  
Ansible can do the following for us:  
1. Configuration management  
2. Application deployment  
3. Task automation  
4. IT orchestration

**Q: Please define what is Ansible Galaxy?**  
Ansible Galaxy refers to the website Galaxy where the users will be able to share all the roles to a CLI ( Command Line interface) where the installation, creation, and managing of roles happen

**Q: Do you know what language Ansible is written in?**  
Ansible is written in Python and PowerShell  
   
**Q: Please explain what is Red Hat Ansible?**  
Ansible and Ansible Tower by Red Hat, both are an end to end complete automation platforms which are capable of providing the following features or functionalities:

1. Provisioning  
2. Deploying applications  
3. Orchestrating workflows  
4. Manage IT systems  
5. Configuration of IT systems  
6. Networks  
7. Applications

All of these activities are dealt by Ansible where it can help the business to solve the real time business problems.

**Q: Is Ansible is an open source tool?**  
Yes, Ansible is an open source tool which is a powerful automation software tool that one can use.

**Q: Why you have to learn Ansible?**  
Ansible is more a tool for servers but does it have anything for networking. If you closely look into it, there is some support available in the market for networking devices. Using this tool, it will give you an overall view of your environment and also the knowledge how it works when it comes to network automation.

It is one of those tools where it is considered to be good to explore a new tool.

**Q: What are Ansible server requirements?**  
If you are a windows user then you need to have a virtual machine in which Linux should be installed.  
It requires Python 2.6 version and higher.

**Q: How can you connect to other devices within Ansible?**  
Once, Ansible is installed and the basic setup has been completed, an inventory is created. This would be the base and one can start testing ansible. To connect to a different device then you have to use “Ping module”. This can be used as a simple connection test.

Ansible - m ping all

**Q: Can you build your own modules with Ansible?**  
Yes, we can create or own modules within Ansible.  
It is an open source tool which primarily works on Python. If you are good at programming in Python you can start creating your own modules in few hours from scratch and you don't need to have any prior knowledge of the same.

**Q: How can you find information in Ansible?**  
After completing the basic setup, one has to make sure to find out the module called “setup” module. Using this setup module, you will be able to find out a lot of information.

**Q: What does Fact mean in Ansible?**  
The term “Facts” is commonly used in Ansible environment. They are described in the playbooks areas where it displays known and discovered variables about the system.  Facts are used to implement conditionals executions and also used for getting ad-hoc information of the information.

You can see all the facts via:

$ ansible all- m setup

So if you want to extract only certain part of the information then you can use “setup” module where you will have an option to filter out the output and just get hold of the fact that you are in need of.

**Q: What is ask\_pass in ansible?**  
 The ask\_pass is a control in Ansible Playbook.  
This controls whether ansible playbook to prompt a password by default. Usually, the default behavior is no:

It is always set to ask\_pass=True

If you are using SSH keys for authentication purposes then you really don’t have to change this setting at all.

**Q: Explain What is ask\_sudo\_pass**  
This control is very similar to ask\_pass  
The ask\_sudo\_pass controls the Ansible Playbook to prompt a sudo password. Usually, the default behavior is no:

ask\_sudo\_pass= True

One has to make sure and change this setting where the sudo passwords are enabled most of the time.

**Q: Explain what is ask\_vault\_pass?**  
Using this control we can determine whether Ansible Playbook should prompt a password for the vault password by default. As usual, the default behavior is no

ask\_vault\_pass= True

**Q: Explain Callback\_plugin in Ansible?**  
Callbacks are explained as a piece of code in ansible environments where get is used call a specific event and permit the notifications.

This is more sort of a developer related feature and allows low-level extensions around ansible so that they can be loaded from different locations without any problem.

**Q: Explain Module utilities in Ansible?**   
Ansible provides a wide variety of module utilities which help the developers while developing their own modules. The basic.py is a module which provides the main entry point for accessing the Ansible library and using those as basics one can start off working.

**Q: Where is the unit testing is available in Ansible?**  
Unit tests for all the modules are available in .test/units/modules.  
Firstly you have to setup your testing environment

**Q: Explain in detail about ad-hoc command?**  
Well, ad-hoc commands is nothing but a command which is used to do something quickly and it is more sort of a one-time use.  Unlike, the playbook is used for a repeated actions which is something that is very useful in Ansible environment. But there might be scenarios where we want to use ad-hoc commands which can simply do the required activity and it is a nonrepetitive activity.

### Difference between a variable name and Environment Variables?

**Answer:**

|  |  |
| --- | --- |
| **Variable Name** | **Environment Variable** |
| Variable name can be built by adding String. | To access the environment variable need to access existing variable. |
| {{ hostvars[inventory\_hostname][‘ansible\_’ + which\_interface][‘ipv4’][‘address’] }} | # … vars: local\_home: “{{ lookup(‘env’,’HOME’) }}” |
| We can add Strings | If we want to add the variable we need to open advance playbooks section. |
| For variable name, we use [IPV4 address](https://www.educba.com/ipv4-vs-ipv6/). | For Remote environment variables, use {{ ansible\_env.SOME\_VARIABLE }} |

**Q1. What Is Ansible ?**  
Ansible is a software tool to deploy application using ssh without sny downtime.It is also used to manage and configure software applications. Ansible is developed by Python language.

**Q2. What is Ansible galaxy?**  
Galaxy refers to bothe website and CLI tool used to the interact with the website where you can download and share roles with other members of the ansible communty.

**Q3. What is Forks in Ansible**  
Forks is a way to improve your ansible performance defining how many ansible processes will be created to communicate with remote hosts.

**Q4. Briefly Explain Pipelining in Ansible.**  
Pipelining allows Ansible to use stream commands over a single connection instead of opening connection for each ansible command.

**Q5. How can we use controlpersist to speed up ansible deployment?**  
This allows us to create a single master connection that can be reused subsequently for a given amount of time.

**Q6. Explain “fire and forget” concept in ansible.**  
This allows us to  run a task without waiting for completion. You simply run the task Async and set poll=0.  Later in the playbook, use async\_status to check the status of the job.

**Q7. How do we make a variable available to a host or group without including it in the inventory file?**  
You can create a variable file under group\_vars. For example, lets say we want to make a variable available too the **webserver** host group, you simple create **group\_vars/webservers** and define the variable inside the file.

**Q8. What’s The Use Of Ansible ?**  
Ansible can be used in IT infrastructure to manage and deploy software applications to remote nodes. For example, let’s say you need to deploy a single software or multiple software to 100’s of nodes by a single command, here ansible comes into picture, with the help of Ansible you can deploy as many as applications to many nodes with one single command, but you must have a little programming knowledge for understanding the ansible scripts.  
We’ve compiled a series on Ansible, title ‘Preparation for the Deployment of your IT Infrastructure with Ansible IT Automation Tool‘, through parts 1-4 and covers the following topics.

**Q9. Why would you want to disable ansible facts.**  
You can disable facts if it its not being used to save on memory used for storing the variables created during facts/

**Q10. What are ansible strategies?**  
Ansible strategies are plugins that modifies the way ansible works. For example, the linear strategy executes task on the host in series waiting for all hosts to complete before moving to next task while free moves on to next task once it finish execution on a host. For debug strategy, it executes linear but triggers debugger on failure.

**Q11. How can I set the PATH or any other environment variable for a task or entire playbook?**  
Setting environment variables can be done with the *environment* keyword. It can be used at the task or the play level:

environment:

PATH: "{{ ansible\_env.PATH }}:/thingy/bin"

SOME: value

Note : starting in 2.0.1 the setup task from gather\_facts also inherits the environment directive from the play, you might need to use the *|default* filter to avoid errors if setting this at play level.

**Q12. How Do I Submit A Change To The Documentation ?**  
Documentation for Ansible is kept in the main project git repository, and complete instructions for contributing can be found in the docs.

**Q13. When Should I Use {{ }} ? Also, How To Interpolate Variables Or Dynamic Variable Names ?**  
A steadfast rule is ‘always use {{ }} except when when:‘. Conditionals are always run through Jinja2 as to resolve the expression, so when: failed\_when: and changed\_when: are always templated and you should avoid adding {{}}.  
In most other cases you should always use the brackets, even if previouslly you could use variables without specifying (like with\_ clauses), as this made it hard to distinguish between an undefined variable and a string.  
Another rule is ‘moustaches don’t stack’. We often see this:

{{ somevar\_{{other\_var}} }}

The above DOES NOT WORK, if you need to use a dynamic variable use the hostvars or vars dictionary as appropriate:

{{ hostvars[inventory\_hostname][‘somevar\_’ + other\_var] }}

**Q14. How do I handle different machines needing different user accounts or ports to log in with?**  
Setting inventory variables in the inventory file is the easiest way.  
Ansible 2.0 has deprecated the “ssh” from ansible\_ssh\_user,  ansible\_ssh\_host, and ansible\_ssh\_port to become ansible\_user, ansible\_host, and ansible\_port.

If you are using a version of Ansible prior to 2.0, you should continue using the older style variables (ansible\_ssh\_\*). These shorter variables are ignored, without warning, in older versions of Ansible.

For instance, suppose these hosts have different usernames and ports:

[webservers]

[asdf.example.com](http://asdf.example.com/) ansible\_port=5000 ansible\_user=alice

[jkl.example.com](http://jkl.example.com/) ansible\_port=5001 ansible\_user=bob

You can also dictate the connection type to be used, if you want:

[testcluster]

localhost ansible\_connection=local

/path/to/chroot1 ansible\_connection=chroot

[foo.example.com](http://foo.example.com/) ansible\_connection=paramiko

You may also wish to keep these in group variables instead, or file them in a group\_vars/<groupname> file. See the rest of the documentation for more information about how to organize variables

**Q15. How To Install Ansible ?**  
The best way to get Ansible for Ubuntu is to add the project’s PPA (personal package archive) to your system.

To do this effectively, we need to install the software-properties-common package, which will give us the ability to work with PPAs easily. (This package was called python-software-properties on older versions of Ubuntu.)

* sudo apt-get update
* sudo apt-get install software-properties-common

**Once the package is installed, we can add the Ansible PPA by typing the following command**

sudo apt-add-repository ppa:ansible/ansible

Press ENTER to accept the PPA addition.

**Next, we need to refresh our system’s package index so that it is aware of the packages available in the PPA. Afterwards, we can install the software:**

* sudo apt-get update
* sudo apt-get install ansible
* We now have all of the software required to administer our servers through Ansible.

**Q16. How Do I Generate Crypted Passwords For The User Module ?**  
mkpasswd –method=sha-512

If this utility is not installed on your system (e.g. you are using OS X) then you can still easily generate these passwords using Python. First, ensure that the Passlib password hashing library is installed.

pip install passlib

Once the library is ready, SHA512 password values can then be generated as follows:  
python -c “from passlib.hash import sha512\_crypt; import getpass; print sha512\_crypt.encrypt(getpass.getpass())”

Use the integrated Hashing filters to generate a hashed version of a password. You shouldn’t put plaintext passwords in your playbook or host\_vars; instead, use Vault to encrypt sensitive data.